

# Estructuras de Datos y Algoritmos

Grados de la Facultad de Informática (UCM)

Examen SEGUNDO CUATRIMESTRE, 31 de mayo de 2019

---

## Normas de realización del examen

1. Debes desarrollar e implementar soluciones para cada uno de los ejercicios, probarlas y entregarlas en el juez automático accesible en la dirección <http://exacrc>.
2. En el juez te identificarás con el nombre de usuario y contraseña que has recibido al comienzo del examen.
3. Escribe tu **nombre y apellidos** en un comentario en la primera línea de cada fichero que subas al juez.
4. Del enlace **Material para descargar** dentro del juez puedes descargar un archivo comprimido que contiene material que puedes utilizar para la realización del examen (transparencias de clase, implementación de las estructuras de datos, una plantilla de código fuente y ficheros de texto con los casos de prueba de cada ejercicio del enunciado).
5. Los ficheros con las implementaciones de las estructuras de datos están instalados en el juez, por lo que no es necesario subirlos como parte de tu solución (y conviene no hacerlo).
6. Los ejercicios están identificados con el nombre del tema de la asignatura en el que habrían aparecido si hubieran sido propuestos como ejercicios durante el curso. Para obtener la máxima puntuación, las soluciones deberán seguir los criterios exigidos a los ejercicios de ese tema durante el curso.
7. Tus soluciones serán evaluadas por el profesor independientemente del veredicto del juez automático. Para ello, el profesor tendrá en cuenta **exclusivamente** el último envío que hayas realizado de cada ejercicio.
8. Al terminar el examen, dirígete al puesto del profesor y rellena con tus datos la hoja de firmas que él tendrá. Muéstrale tu documento de identificación.

*Primero resuelve el problema. Entonces, escribe el código.*

— John Johnson

*Comentar el código es como limpiar el cuarto de baño;  
nadie quiere hacerlo, pero el resultado es siempre  
una experiencia más agradable para uno mismo y sus invitados.*

— Ryan Campbell

## Ejercicio 1. Tipos de datos lineales (3 puntos)

Estamos programando un *bot* de Telegram que enviará noticias a sus seguidores y ahora nos toca tratar el escabroso tema de los accidentes de aviación. Cuando se produce uno de estos accidentes, suelen hacerse comentarios del estilo “*Este es el accidente más grave desde febrero de 1995*”.

Nos han pasado un listado (ordenado cronológicamente) de accidentes ocurridos en el pasado y queremos estar preparados para que, cuando ocurra el siguiente accidente, podamos producir un comentario como el anterior.

De hecho, para probar esta funcionalidad, queremos saber cuál habría sido el comentario cuando se produjo cada uno de los accidentes conocidos.

### Entrada

La entrada está formada por una serie de casos. Cada caso comienza con el número  $N$  de accidentes conocidos (un número entre 1 y 250.000). A continuación aparecen  $N$  líneas con la descripción de cada uno de ellos: una fecha (con formato DD/MM/AAAA) y el número de víctimas en ese accidente. Todas las fechas son distintas y el listado está ordenado cronológicamente de menor a mayor.

### Salida

Para cada caso se escribirán  $N$  líneas. La  $i$ -ésima línea contendrá la fecha del último accidente anterior que tuvo (estrictamente) más víctimas que el accidente  $i$ -ésimo. Si no existe tal accidente (en particular, eso ocurre siempre para el primero), se escribirá **NO HAY** en su lugar.

Después de cada caso se escribirá una línea con tres guiones (---).

### Entrada de ejemplo

```
6
19/12/1990 50
01/02/2000 80
10/05/2001 30
20/10/2005 10
08/07/2007 60
10/07/2007 40
```

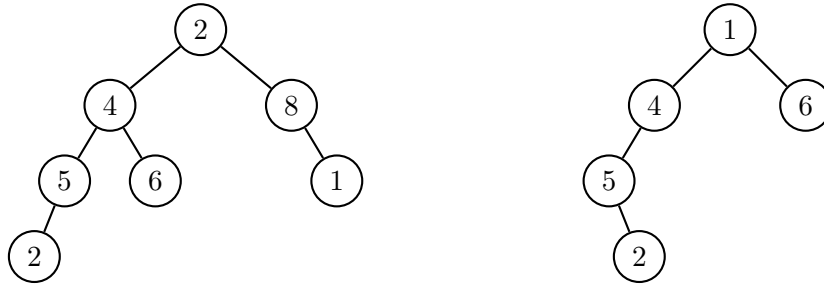
### Salida de ejemplo

```
NO HAY
NO HAY
01/02/2000
10/05/2001
01/02/2000
08/07/2007
---
```

## Ejercicio 2. Tipos de datos arborescentes (3 puntos)

Definimos un *camino* entre dos nodos de un árbol binario como una secuencia de nodos  $n_1 n_2 \dots n_k$  sin repeticiones (por cada nodo del árbol se pasa como mucho una vez) tal que para todo par de nodos consecutivos  $n_i n_{i+1}$  ( $1 \leq i < k$ ) uno de ellos siempre es padre del otro ( $n_i$  es padre de  $n_{i+1}$  o  $n_{i+1}$  es padre de  $n_i$ ). Definimos la *longitud* de un camino  $n_1 n_2 \dots n_k$  como el número de nodos que lo forman,  $k$ .

Dado un árbol binario de números enteros positivos queremos calcular la longitud del camino más largo formado todo él por números pares. Por ejemplo, el siguiente árbol de la izquierda tiene el camino 6 4 2 8 de longitud 4. En cambio, en el árbol de la derecha todos los caminos formados únicamente por números pares tienen longitud 1.



### Entrada

La entrada comienza indicando el número de casos de prueba que vendrán a continuación. Cada caso consiste en una serie de números enteros positivos con la descripción de un árbol binario: el árbol vacío se representa con el valor -1; un árbol no vacío se representa con el valor de un nodo (que denota la raíz), seguido primero de la descripción del hijo izquierdo y después de la descripción del hijo derecho.

### Salida

Para cada caso, se escribirá la longitud del camino más largo entre dos nodos del árbol, formado todo él por números pares.

### Entrada de ejemplo

```
3
2 4 5 2 -1 -1 -1 6 -1 -1 8 -1 1 -1 -1
1 4 5 -1 2 -1 -1 -1 6 -1 -1
3 4 2 -1 -1 6 -1 8 -1 -1 5 4 -1 2 -1 -1 -1
```

### Salida de ejemplo

```
4
1
4
```

### Ejercicio 3. Diccionarios (4 puntos)

Tenemos una lista de las películas emitidas durante el último año en una cadena de televisión. De cada película conocemos los actores que intervienen en ella y el tiempo que aparecen en pantalla durante la película. Queremos obtener la película y el actor *preferidos* por la cadena. La película *preferida* es aquella que más veces se ha emitido, mientras que el actor *preferido* es aquel que más minutos aparece en pantalla contando todas las emisiones. Si existen varios actores que han aparecido el mismo tiempo máximo mostraremos todos ellos por orden alfabético. Si existen varias películas que se han emitido el mayor número de veces, mostraremos la que se ha emitido más recientemente de todas ellas.

#### Entrada

La entrada consta de una serie de casos de prueba. Cada caso se muestra en varias líneas. En la primera se indica el número  $P$  de películas distintas que se han emitido. A continuación se muestra, para cada película, su título y el número  $A$  de actores que aparecen en ella. En la línea siguiente se muestra el nombre de cada actor y su tiempo de actuación (un número positivo) en esta película, separados por blancos. A continuación se muestra el número  $E$  de películas emitidas por la cadena de televisión en todo el año, seguido de los títulos de las películas en el orden en que fueron emitidas. La entrada finaliza con una línea con cero películas.

Los títulos de las películas y los nombres de los actores son cadenas de caracteres sin espacios en blanco. Los títulos y los nombres de actores están escritos siempre de la misma manera.

#### Salida

La salida de cada caso se escribirá en dos líneas. En la primera se muestra el máximo número de veces que se ha emitido una película seguido del título de la película. Si existen varias películas que se han emitido el mismo número máximo de veces, se muestra el título de la última emitida. En la segunda línea se muestra el máximo tiempo que ha aparecido un actor en pantalla seguido de los nombres de los actores que han aparecido ese tiempo máximo en orden alfabético.

#### Entrada de ejemplo

```
2
pelicula1 3
actor1 10 actor2 30 actor3 5
pelicula2 2
actor1 10 actor4 30
3
pelicula2 pelicula2 pelicula1
4
pelicula1 3
actor3 5 actor1 10 actor2 20
pelicula3 1
actor5 40
pelicula4 1
actor1 40
pelicula2 2
actor1 10 actor4 15
5
pelicula2 pelicula1 pelicula2 pelicula3 pelicula1
0
```

### Salida de ejemplo

```
2 pelicula2
60 actor4
2 pelicula1
40 actor1 actor2 actor5
```